Newcastle University

# Newcastle University ePrints

**Emms M, Arief B, Little N, van Moorsel A. Risks of Offline Verify PIN on Contactless Cards. *In:* Ahmad-Reza Sadeghi, ed. *Financial Cryptography and Data Security*. Berlin: Springer Berlin Heidelberg, 2013, pp.313-321.**

**Copyright:**

The final publication is available at Springer via http://dx.doi.org/10.1007/978-3-642-39884-1_26

**Further information on publisher website:** http://www.springer.com/

**Date deposited:** 24th September 2014

**Version of chapter:** Accepted

This work is licensed under a Creative Commons Attribution-NonCommercial 3.0 Unported License

**ePrints – Newcastle University ePrints**
**http://eprint.ncl.ac.uk**

# Risks of Offline Verify PIN on Contactless Cards

Martin Emms, Budi Arief, Nicholas Little, and Aad van Moorsel

School of Computing Science, Newcastle University, Newcastle upon Tyne, UK
{martin.emms,budi.arief,n.little,aad.vanmoorsel}@ncl.ac.uk

**Abstract.** Contactless card payments are being introduced around the world allowing customers to use a card to pay for small purchases by simply placing the card onto the Point of Sale terminal. Contactless transactions do not require verification of the cardholder's PIN. However our research has found the redundant verify PIN functionality is present on the most commonly issued contactless credit and debit cards currently in circulation in the UK. This paper presents a plausible attack scenario which exploits contactless verify PIN to give unlimited attempts to guess the cardholder's PIN without their knowledge. It also gives experimental data to demonstrate the practical viability of the attack as well as references to support our argument that contactless verify PIN is redundant functionality which compromises the security of payment cards and the cardholder.

**Keywords.** Contactless Payments, Verify PIN, NFC, EMV, Chip & PIN, Credit Card, Debit Card, Card Payment.

## 1 Introduction

The EMV[1] specifications [5][6] control the operation of 1.62 billion of payment cards and 23.8 million of Point of Sale terminals worldwide [15]. EMV payments can be contact transactions commonly termed Chip & PIN or contactless transactions also known as Near Field Communication (NFC).

Contact payments require the cardholder to insert their card into the Point of Sale terminal and enter their PIN to authorise the transaction. Contact transactions can be any value up to the card limit or available balance on the card. Contactless payments are designed to be a convenient way to pay for low value transactions (currently up to £20 per transaction in the UK) with a card rather than cash. Designed to be faster than a traditional Chip & PIN transaction, the card is simply placed in close proximity (approximately 4cm) to the Point of Sale terminal to authorise the payment, PIN entry is not required.

In the UK the EMV specification for contact transactions supports PIN verification locally by the card (*offline*) and PIN verification remotely by the bank's computers (*online*). The specifications for contactless transactions specifically exclude the use of *offline* PIN verification (full details in [6] Book A section 5.9.3 and [10] section 2.4 point 5). Contact-

[1] Europay, MasterCard, Visa is a collaboration between Visa, MasterCard, American Express and JCB to create an interoperable card payment system.

less *offline* PIN verification requires the PIN to be transmitted wirelessly to the card which poses a security risk from eavesdropping.

The EMV specification only permits PIN entry in contactless transactions made using NFC enabled mobile devices. PIN entry is not permitted for contactless card transactions. Mobile device payments are controlled by Consumer Device CVM[2] rules, which permit *online* PIN verification, but not *offline* PIN (full details in [6] Book C3 sections 2.1 and 5.7).

This paper examines the security implications of the verify PIN functionality intended for Chip & PIN operation also being available over the contactless interface, where it can be accessed without the cardholder's knowledge or consent. Surprisingly many of the contactless cards currently in circulation in the UK allow access to *offline* verify PIN.

The attack scenario presented draws upon research carried out into the predictability of PINs [2] which shows that there is a subset of PINs that are much more commonly used; meaning guesses from this subset are much more likely to be successful.

The implementation work builds upon related investigations into the vulnerability of EMV contactless payment cards to various attacks, such as skimming [7][8] and transaction relay [4][9]. These papers show that the wireless interface makes contactless payment cards vulnerable to new modes of attack that were not present in Chip & PIN. Other research [3][11] show that the EMV protocol sequence can be manipulated to produce erroneous behaviour in the cards and the Point of Sale terminals.

In what follows, we first introduce the attack scenario then the technology used and finally the performance results demonstrating the practicality of the attack. A critical part of our software implementation is the ability to find and attack EMV payment cards contained in a wallet with various other contactless cards. Our software implements the ISO-14443 part 3 protocol sequence for card initialisation and anti-collision. It can identify multiple cards, select each card in turn and communicate with each card once selected.

## 2 Attack Scenario

The attack scenario outlined in this paper is presented as supporting evidence of our assertion that allowing contactless access to *offline* verify PIN represents a tangible threat to a large number of EMV payment cards currently in circulation in the UK.

Newcastle University, like many other companies and institutions, uses NFC enabled identity cards to control access to our buildings. When entering the building, many of us place our whole wallet on the door access reader as it is quicker and easier than taking the access card out of the wallet. This gives an attacker the opportunity to access the other cards in the wallet, communicating with any contactless payment cards also present.

Given that the person will enter the building on a regular basis and that the number of available PIN attempts is reset each time the payment card is used in a Point of Sale terminal or ATM, the attacker can have unlimited attempts to guess a card's PIN.

In our experimental implementation of the attack scenario we make use of (i) a protocol sequence which exploits the verify PIN functionality (ii) the ability to access multiple cards in a single wallet presented to the door access reader (iii) a strategy for guessing PINs [2] which will yield greatest number of correct guesses.

[2] Cardholder Verification Method is used to approve the transaction either by PIN or by signature.

## 2.1 PIN Verify Protocol Sequence

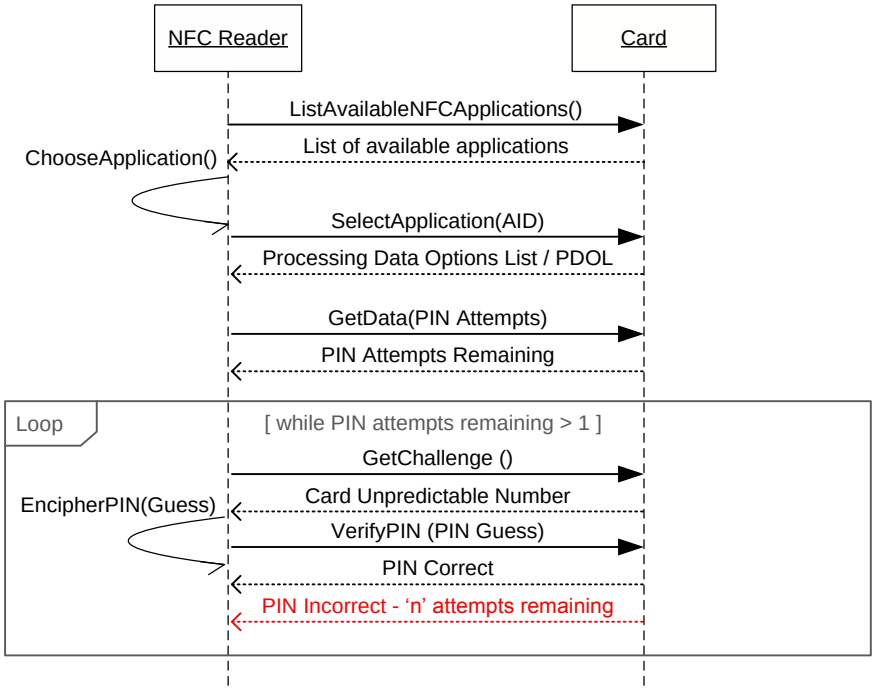


**Fig 1 -** Verify PIN protocol sequence

The full protocol sequence (Fig 1) is designed to guess the PIN without locking the card. Locking occurs when all of the available PIN attempts are used (i.e. the card is locked when the counter for PIN attempts remaining becomes zero). The protocol uses the minimum number of commands possible so that it can be completed quickly (<500ms) to avoid arousing the suspicions of the cardholder. Moreover, to avoid locking the card we need to keep at least one PIN attempt remaining on the card. The protocol sequence is therefore limited to a maximum of two guesses each time the cardholder uses the door. However, over time the attack has multiple chances to run the protocol sequence as the person will regularly return to the door access reader and each time the card is used in a Point of Sale terminal or ATM, the PIN attempt counter is reset, giving more chances for further guesses.

The PIN verify protocol sequence described above ensures that at least one PIN attempt is left on the card. However the logic can be changed to create a nuisance attack which wipes out all of the available PIN attempts on all of the EMV payment cards in the wallet. This would not yield any financial gain, but there are many malicious attacks performed purely for the nuisance value. A card that has zero PIN attempts remaining cannot be reactivated at the Point of Sale terminal and the cardholder must to go to a bank ATM.

### 2.2 Reading Multiple Cards

The scenario requires reader software capable of distinguishing between multiple NFC cards in a wallet, allowing it to locate the EMV payment cards (implementation details can be found in section 3.2). This also gives the potential to look for additional data such as the cardholder's birthday on the other cards in the wallet, such as loyalty cards which may hold personal data unencrypted. Bonneau et al. [2] shows that knowing the person's birthday increases the chances of guessing their PIN within 6 guesses from 1.94% to 8.23%.

### 2.3 PIN Guessing Strategy

The attack scenario presented accesses the card each time the cardholder enters the building. This gives it potentially unlimited guesses at the PIN over time, two guesses each time the door access is used. Bonneau et al. [2] presents a survey containing a study of 1,351 respondents, 805 of which detailed the respondents' choice of PIN and their reason for choosing it. The survey shows that 23% of respondents chose a memorable date (birthday and anniversary) as their PIN. The paper goes further and identifies a list of PINs which are statistically more likely; using this list, the paper calculates that given 6 guesses, the chance of correctly guessing the PIN is 1.94%, which rises to 8.23% if the birthday of the cardholder is known. This research is backed up by a recent news story [14] where a burglar stole a wallet in which he found a driving licence and two ATM cards, he correctly guessed the PIN from the date of birth on the driving licence and was able to obtain £1,000 from a nearby ATM.

## 3 Software Implementation

The experimental work in preparing this paper includes (i) an implementation of the verify PIN protocol sequence which makes multiple attempts to guess the PIN of any EMV payment card detected in the wallet (ii) a multiple card reader implementation which will identify and communicate with all of the contactless cards in the wallet.

The experiments were performed using an ACR122-U contactless card reader [1] and the Java™ Smart Card I/O API [13].

### 3.1 Verify PIN Implementation

The UML sequence diagram (Fig 1) illustrates the protocol sequence required to perform the verify PIN attack sequence. The sequence employs the minimum number of commands which achieve two contactless verify PIN attempts, this minimises total execution time (on average 457.2ms) for the sequence. Minimising execution time is important to ensure that the attack is not easily detected by the cardholders using the door access.

The protocol sequence is initiated when the multiple card reader (section 3.2) detects an EMV payment card in the wallet. The protocol sequence therefore starts with the EMV payment card in the `active` state ready to accept commands (see Table 1 for a full explanation of the possible card states). Once the reader has established communication with the card, it reads the number of PIN attempts remaining using `GetData(PIN Attempts)`. It then calls the verify PIN command in a loop. The card responds with 0x9000 if the PIN is

correct or 0x63Cn if the PIN is incorrect, where 'n' is the number of PIN attempts remaining. The loop is repeated until the correct PIN is guessed or only one PIN attempt remains.

We observed that the contactless PIN is the same as the contact PIN, this was confirmed by changing the card's contact PIN using an ATM and verifying that the contactless PIN had also changed.

### 3.2 Multiple Card Reader Implementation

EMV contactless payment cards are compliant with the ISO-14443:Part 3 which defines the disambiguation and activation sequence. Disambiguation involves obtaining the Unique Identifiers (UID) of each of the cards in the NFC field. Once this is complete, the UID is used to activate each card individually. The card is then `ready` to accept commands. For successful communication only one card can be `active` at any one time. Table 1 below describes the transitions between the different states `idle, ready, active` and `halt` which allow the reader to successfully communicate with an individual card when there are multiple cards in the field.

**Table 1.** ISO-14443 Card State Transitions

| State | Description |
|---|---|
| `idle` | Upon entry to the NFC field all cards will power up into the `idle` state. |
| `ready` | The reader transmits `REQA` / `WUPA` command putting the cards into the `ready` state. Once all of the cards are in the `ready` state the anti-collision loop sequence can begin. |
| `active` | The anti-collision loop sequence is an iterative process used by the NFC reader to find the UID of the next card in the field. The anti-collision command is repeatedly sent to all cards until only one card answers with a complete UID and no collisions. The UID is then used in the `Select` command which moves that card into the `active` state. At this point the reader can communicate with the card using the card type specific protocol (EMV, MIFARE etc.) or instruct the card to `halt` and store the UID for future use. |
| `halt` | To communicate with the next card in the NFC field the reader must `halt` the currently `active` card. Cards can be re-awakened from the `halt` state using the `WUPA`. |

The process of communicating with multiple cards is as follows:

1. the anti-collision loop finds the UID of each card in turn
2. Select(UID) moves the card with the given UID into the `active` state
3. the `active` card is now ready for communication with the reader, only one card at a time can be `active`
4. `halt` is used to stop communicating with the card and move to the next card

The current implementation of the disambiguation and activation sequence is compatible with all ISO-14443:Part 3 compliant cards. Once disambiguation is complete each card type has its own specific communication protocol. We have implemented protocol sequences for three commonly available card types: EMV payment cards, MIFARE classic door access cards and MIFARE DESFire travel pass cards. Communication with the implemented card

types is not affected if an unknown card type is also present in the NFC field, the unknown card type is simply ignored once the disambiguation process has identified its UID. The software utilises hardware commands specific to the NXP PN532 chipset [12] to perform the anti-collision loop, disambiguation and card selection.

# 4 Results

The test results in this section focus on the time taken to perform each of the steps involved in performing the attack scenario presented in section 2. These results are presented to support our assertion that the delay introduced by the attack would not arouse the suspicions of the users of the door access system.

## 4.1 Verify PIN protocol sequence

Based on the data obtained in our tests the average time taken to perform the complete protocol sequence (Fig 1) was only 457.2ms; thereby strengthening the case that the door access reader attack scenario can be implemented without raising the suspicions of the users of the door access system.

The time taken to perform each of the commands in the verify PIN protocol sequence is detailed in Table 2, which shows the average time and standard deviation calculated from 20 test runs performed using EMV payment cards issued by a UK bank.

**Table 2.** Verify PIN command execution times

| Command | average (ms) | standard deviation (sec) |
|---|---|---|
| `ListAvailableNFCApplications()` | 18.4 | 12.7 |
| `SelectApplication(AID)` | 19.2 | 5.5 |
| `GetData(PIN attempts)` | 29.8 | 17.9 |
| `GetChallenge()` | 24.6 | 7.0 |
| `VerifyPIN(incorrect PIN)` | 175.8 | 7.2 |
| `GetChallenge()` | 12.2 | 6.8 |
| `VerifyPIN(correct PIN)` | 177.2 | 9.6 |
| `Complete Protocol Sequence` | 457.2 | 24.9 |

The results show that 77.2% of the total time was taken by the card responding to the `VerifyPIN()` command. It is also interesting to note that there is no significant difference between a correct PIN (177.2ms) and an incorrect PIN (175.8ms).

## 4.2 Multiple Card Identification

For the multiple card identification tests we used three of the more popular contactless card types: EMV payment cards, MIFARE classic door access cards and MIFARE DESFire travel pass cards. The test results in Table 3 show the average time (over 60 test runs) to

identify each card, when there are multiple cards in the NFC field. Results of the tests show the identification of each card takes longer when more cards are in the field.

**Table 3.** Multiple Card Identification Times

| Cards in NFC Field | 2 cards | 3 cards | 4 cards | 5 cards |
|---|---|---|---|---|
| Identification of Each Card (ms) | 214.36 | 285.82 | 305.95 | 358.30 |
| Standard Deviation (ms) | 16.91 | 16.66 | 72.54 | 53.87 |

The maximum number of cards that the ACR-122U reader (used in our tests) can identify in the NFC field varies by card type. Table 4 shows the maximum number of each card type that the reader could identify and communicate with. The first three rows show tests with a single card type in the NFC field. The following three rows represent wallets containing a mixture of card types, with at least one EMV payment card and one MIFARE classic door access card (as the attack scenario described is based on wallets containing these two cards).

**Table 4.** Maximum Cards in NFC field

| | EMV payment | MIFARE classic | MIFARE DESFire |
|---|---|---|---|
| Single card type | 2 cards | 5 cards | 4 cards |
| Multiple card types | 2 cards<br>1 card<br>1 card | 1 card<br>1 card<br>3 cards | 1 card |

### 4.3 Total Attack Time

Table 5 illustrates the total time taken by the verify PIN attack on two example wallets: *wallet 1* containing one MIFARE classic door access card and one EMV payment card; and *wallet 2* containing one MIFARE classic, one EMV and one MIFARE DESFire travel pass. The complete sequence identifies all of the cards present in the wallet and then performs two PIN guesses on the EMV payment card.

**Table 5.** Multiple Card Identification and Communication Time

| Scenario | Identify Card (ms) | Communication (ms) | Total (ms) |
|---|---|---|---|
| *wallet 1* | 428.73 | 457.20 | 855.93 |
| *wallet 2* | 643.09 | 457.20 | 1070.29 |

In summary, the test results (Table 3) show that it is possible to attack a wallet containing multiple card types. Moreover, Table 5 shows that for both *wallet 1* and *wallet 2*, the total attack time of around 1 second is fast enough to avoid detection by the cardholder. The attack should also delay the green light that signifies the card has been read and delay the opening of the door. This will reassure the cardholder that the system is operating normally (if a little slowly) and allows time for the attack to complete.

## 5 Conclusion

The attack scenario described in this paper exploits contactless verify PIN to give potentially unlimited attempts to guess the cardholder's PIN without their knowledge, this significantly increases the odds that the attack will guess their PIN correctly. The implementation work has successfully built and tested software that proves this attack scenario is technically viable. The timing tests prove that the attack protocol sequence can be performed in less than 1 second (*wallet 1*), making it possible to access the payment cards in the wallet without arousing the suspicions of the cardholder.

It is our assertion that the attack scenario and experimental implementation work presented in this paper make a compelling case that contactless verify PIN can be misused to find out the PIN of the card without the knowledge of the cardholder. This significantly impacts the underlying security assumption of the Chip & PIN payment system, that an attacker can only gain knowledge of the cardholder's PIN through the negligence or collaboration of the cardholder. Moreover, offline verify PIN is not required in the processing of contactless transactions and is therefore redundant functionality. These findings suggest that it would be prudent to remove the contactless verify PIN functionality. It would also help to educate cardholders remove their card from their wallet before placing it on a reader.

## References


1. Advanced Card Systems: ACR122U NFC Reader Application Programming Interface. http://www.acs.com.hk/drivers/eng/API_ACR122U_v2.00.pdf. Accessed 2013-01-29 (2011)
2. Bonneau J. Preibusch S., Anderson R.: A birthday present every eleven wallets? The security of customer-chosen banking PINs. International Conference on Financial Cryptography (2012)
3. Choudary, O.S.: The Smart Card Detective: a hand-held EMV interceptor. Cambridge (2010)
4. Drimer, S. and Murdoch, S.: Keep Your Enemies Close: Distance Bounding Against Smartcard Relay Attacks. USENIX Security Symposium (2006)
5. EMVCo. EMV Specifications for Payment Systems, Books 1,2,3 and 4, Version 4.3 (2011)
6. EMVCo. EMV Contactless Specifications for Payment Systems, Books A,B,C-1,C-2,C-3,C-4 and D, Version 2.2 (2012)
7. Emms, M.: Practical Attack on Contactless Payment Cards. HCI2011 Workshop - Heath, Wealth and Identity Theft (2011)
8. Francis, L., Hancke, G., Mayes, K., Markantonakis, K.: Potential Misuse of NFC Enabled Mobile Phones with Embedded Security Elements as Contactless Attack Platforms. International Conference for Internet Technology and Secured Transactions (2009)
9. Francis, L., Hancke, G., Mayes, K., Markantonakis, K.: Practical Relay Attack on Contactless Transactions by Using NFC Mobile Phones (2011)
10. MasterCard: PayPass - M/Chip Acquirer Implementation Requirements (2006)
11. Murdoch, S., Drimer, S., Anderson, R., Bond, M.: Chip and PIN is Broken. IEEE Symposium on Security and Privacy (2010)
12. NXP PN532 User Manual (2007 http://www.adafruit.com/datasheets/pn532um.pdf. Accessed 2013-01-29 (2007)
13. Oracle: Java Smart Card I/O API. http://docs.oracle.com/javase/7/docs/jre/api/security/smartcardio/spec/javax/smartcardio/package-summary.html. Accessed 2013-01-29 (2012)
14. Willey, G.: PIN Number burglar used victims' card. Newcastle Evening Chronicle 2012-04-27
15. Worldwide EMV Deployment. http://www.emvco.com/about_emvco.aspx?id=202. Accessed: 2013-01-29 (2011)